Homework3 ： Hand-written Character Recognition

Problem Description

Computer Vision is an important research direction and application area of data analysis and artificial intelligence, and has received wide attention from researchers and industry. Especially in recent years, with the development of deep learning technology, deep models have achieved better results than humans in several computer vision tasks, and technologies such as "face payment" have gradually become popular in our daily life. In this third assignment, we will explore a common but important computer vision problem using deep learning methods: handwritten character recognition, i.e., given some images of handwritten characters and their labels, we want to train a deep learning model that can predict the corresponding characters for a new image of handwritten character.

Data Description

In this assignment, we will use the Omniglot (https://github.com/brendenlake/omniglot) [1][2]. This dataset contains a total of 1623 characters from 50 different alphabet systems. For each of these 1623 characters, the dataset collects handwritten images of the character written by 20 different people. In machine learning terms, this dataset can be viewed as consisting of a total of 1623*20=32460 samples, which are evenly divided into 1623 classes.

Unlike the traditional handwritten character recognition dataset MNIST (which recognizes handwritten digits from 0-9), Omniglot has more sample classes and covers different alphabet systems, making this dataset more practical and challenging for both research and application. In addition, although this dataset has many categories, the size of each image is small (each handwritten image is represented as 28x28 black and white pixel points in the dataset we used), so this dataset requires less computational resources than real image classification datasets such as CIFAR-10 and ImageNet (e.g., it is more suitable to use when GPU computation is not available).

All data are stored in the omniglot_resized folder, which has 50 subfolders corresponding to 50 different symbol systems, and several folders within each symbol system folder corresponding to the character set of that symbol system, each of which contains 20 images. We provide a function in utils.py to read the pictures and divide the training/test dataset for your reference (of course, you can also write your own function for reading).

[1] Human-level concept learning through probabilistic program induction. Science, 2015.

[2] The Omniglot Challenge: A 3-Year Progress Report. arXiv:1902.03477.

Experimental Requirements

In this experiment, we would like you to explore the application of deep learning algorithms about handwritten character recognition data. The following are some specific requirements.

1 Pre-defined experimental environment: 50 randomly selected categories from all categories for classification, using 15 images in each category as training data and 5 images as test data. It is required to compare (a) a fully connected neural network with at least one hidden layer; (b) a convolutional neural network (CNN) based model that can be an existing architecture or designed by yourself. Other hyperparameters (e.g., optimizer, learning rate, number of iterations, etc.) can be set by yourself. Report the experimental results and compare the advantages and disadvantages of the different approaches.

2 Do some trials and comparison, including but not limited to the following aspects.

2.1 How does the split of the dataset, e.g. number of categories, number of training/testing samples, etc., influence the experimental results?

2.2 Influence of models, e.g., how do different architectures influence the experimental results ? like fully connected networks, convolutional neural network architectures.

2.3 Influence of hyperparameters: How do different hyperparameters (e.g. initialization method, optimizer, number of iterations, learning rate, etc.) affect the results?

2.4 If not used in experiment (1), try some deep learning techniques such as dropout, batch normalization, etc.

Note: This part is more open, and we encourage you to try more. There is no fixed standard answer, you can write down what you think is reasonable.

Report Requirement

The following should be included in the report.

1 (60 points) The two model structures and other experimental settings used in experimental requirement (1), report the experimental results, and compare the advantages and disadvantages of different methods.

As a baseline, we tried a simple CNN (with the following structure) that can achieve about 85% accuracy (this result is for reference only and is not required to necessarily achieve this result)

2 (30 points) Try and compare at least three aspects of the experimental requirements

(2), report the results and make some necessary discussion.

3 (10 points) Other thoughts and feelings, etc.

- Conv2D: filters = 32; kernel_size = (3, 3); padding='valid'
- BatchNormalization
- Relu
- Conv2D: filters = 32; kernel_size = (3, 3); padding='valid'
- BatchNormalization
- Relu
- MaxPooling2D: pool_size = (2,2)

- Conv2D: filters = 64; kernel_size = (3, 3); padding='valid'
- BatchNormalization
- Relu
- Conv2D: filters = 64; kernel_size = (3, 3); padding='valid'
- BatchNormalization
- Relu
- MaxPooling2D: pool_size = (2,2)

- Dense: 512
- BatchNormalization
- Relu
- Dropout
- Dense: 50
- Softmax

A baseline CNN architecture

Submission Requirement

1 The report can be in Chinese or English and should not exceed 10 pages of A4 paper.

2 Please put all codes in one file. Note that we do not execute everyone's code, but only use it as a check. Do not submit any intermediate results or raw data.

3 All submissions should be packaged into a zip file for uploading. The zip file name, as well as all reports and codes, should be named after the academic number, e.g. 2020001002.pdf, 2020001002.py, 2020001002.zip. Please check carefully (as we will be using a code checking software for checking whether your code is the same with others)

4 If the above submission requirements are not met, 5% of the points will be deducted.

5 Late submission criteria: 20% of the score will be subtracted for each week of late submission.

6 The assignment is to be completed independently. It is strictly forbidden to copy others' code, results, reports, or any other materials, including online or open source content, such as code on github or publicly available code from the authors of the paper. However, we allow the use of self-contained functions in software, such as various libraries in matlab, python, or functions, layers, auto derivatives, etc. in machine learning frameworks such as pytorch, tensorflow, keras, etc. Please note that copying other people's results and actively providing your homework for others to copy are considered academic misconduct (so, please do not send code, reports, etc. to others). Academic misconduct will result in 0 points for that assignment, and/or failure of this course, and/or other penalties as stipulated in the Rules for the Implementation of the Regulations on Student Disciplinary Action of Tsinghua University.

7 Academic misconduct is a red line, and we will strictly enforce it, so be sure to pay attention!